

Practice Question User Feedback

Controllers

FeedbackController

```
package com.demo.controllers;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;
import com.demo.entities.Feedback;
import com.demo.services.FeedbackService;
@RestController
public class FeedbackController {
    @Autowired
    FeedbackService feedbackService;

    @GetMapping(value="/feedback")
    public Iterable<Feedback> getAllFeedbacks() {
        return feedbackService.GetAllFeedback();
    }

    @PostMapping(path="/feedback", consumes=
{MediaType.APPLICATION_JSON_VALUE})
    public Feedback addNewFeedback(@RequestBody Feedback fb) {
        Feedback newFb = new Feedback(fb.getComments(), fb.getRating(),
fb.getUser());
        feedbackService.addNewFeedback(newFb);
        return newFb;
    }
}
```

TestFormcontroller

```
package com.demo.controllers;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import com.demo.entities.Feedback;
```

```

import com.demo.services.FeedbackService;
@Controller
public class TestFormController {
    @Autowired
    FeedbackService feedbackService;

    @GetMapping(value= "/test_form")
    public String showTestForm(ModelMap model) {
        model.addAttribute("test", new Feedback());
        return "testform";
    }

    @PostMapping(value="/test_form")
    public String submitTestForm(@ModelAttribute("testUser") Feedback fb,
ModelMap m) {
        feedbackService.addNewFeedback(fb);
        m.addAttribute("test", fb);
        return "post";
    }
}

```

Entity

Feedback

```

package com.demo.entities;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import lombok.Data;
@Entity
@Data
public class Feedback {
    @Id
    private int id;
    @Column(name="comments")
    private String comments;
    @Column(name="rating")
    private int rating;
    @Column(name="user")
    private String user;
}

```

```

    public Feedback() {
        super();
    }
    public Feedback(String comments, Integer rating, String user) {
        this.comments = comments;
        this.rating = rating;
        this.user = user;
    }
    public String getComments() {
        return comments;
    }
    public void setComments(String comments) {
        this.comments = comments;
    }
    public Integer getRating() {
        return rating;
    }
    public void setRating(Integer rating) {
        this.rating = rating;
    }
    public String getUser() {
        return user;
    }
    public void setUser(String user) {
        this.user = user;
    }

    @Override
    public String toString() {
        return "Feedback [id=" + id + ", comments=" + comments + ",
rating=" + rating + ", user=" + user + "]";
    }
}

```

FeedbackService

FeedbackService

```
package com.demo.services;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import com.demo.entities.Feedback;
import com.demo.repositories.FeedbackRepository;
```

```

@Service
public class FeedbackService {

    @Autowired
    FeedbackRepository feedbackRepo;

    public Iterable<Feedback> GetAllFeedback() {
        return feedbackRepo.findAll();
    }

    public Feedback addNewFeedback(Feedback fb) {
        return feedbackRepo.save(fb);
    }
}

```

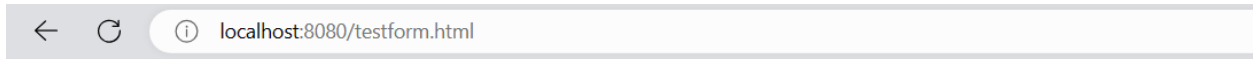
FeedbackRepository

FeedbackRepository

```

package com.demo.repositories;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;
import com.demo.entities.Feedback;
@Repository
public interface FeedbackRepository extends CrudRepository<Feedback, Integer> {
    public Feedback findByUser(String feedback);
}

```

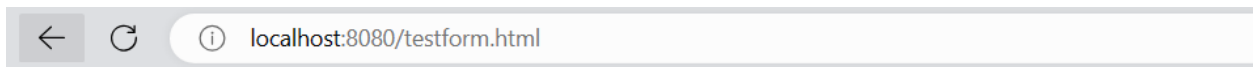


Feedback Test Form

User:

Comments:

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".



Feedback Test Form

User:

Comments:

If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".

http://localhost:8080/index.html?user=Yash&comments=Excellent

Landing Page

[Test Form](#)

[See all Feedbacks](#)

Can only use these link below if you have the jersey dependency added to this dependency. Jersey has been added to this project so it can use the links below.

[See all feedbacks as Json format](#)

[See Json's in profile](#)

localhost:8080/feedback

```
1  [  
2    {  
3      "comments": "Excellent",  
4      "rating": 5,  
5      "user": "Yash"  
6    },  
7    {  
8      "comments": "Good",  
9      "rating": 5,  
10     "user": "Jhon"  
11   },  
12   {  
13     "comments": "Very Good",  
14     "rating": 4,  
15     "user": "MAX"  
16   }  
17 ]
```

```

1  {
2    "_embedded": {
3      "feedbacks": [
4        {
5          "comments": "Excellent",
6          "rating": 5,
7          "user": "Yash",
8          "_links": {
9            "self": {
10              "href": "http://localhost:8080/feedbacks/1"
11            },
12            "feedback": {
13              "href": "http://localhost:8080/feedbacks/1"
14            }
15          }
16        },
17        {
18          "comments": "Good",
19          "rating": 5,
20          "user": "Jhon",
21          "_links": {
22            "self": {
23              "href": "http://localhost:8080/feedbacks/1001"
24            },
25            "feedback": {
26              "href": "http://localhost:8080/feedbacks/1001"
27            }
28          }
29        },
30        {
31          "comments": "Very Good",
32          "rating": 4,
33          "user": "MAX",
34          "_links": {
35            "self": {
36              "href": "http://localhost:8080/feedbacks/1002"
37            },
38            "feedback": {
39              "href": "http://localhost:8080/feedbacks/1002"
40            }
41          }
42        }
43      ]
44    },
45    "_links": {
46      "self": {
47        "href": "http://localhost:8080/feedbacks"
48      },
49      "profile": {
50        "href": "http://localhost:8080/profile/feedbacks"
51      },
52      "search": {
53        "href": "http://localhost:8080/feedbacks/search"
54      }
55    }
56  }

```



localhost:8080/feedbacks/1002

```
1 {  
2   "comments": "Very Good",  
3   "rating": 4,  
4   "user": "MAX",  
5   "_links": {  
6     "self": {  
7       "href": "http://localhost:8080/feedbacks/1002"  
8     },  
9     "feedback": {  
10      "href": "http://localhost:8080/feedbacks/1002"  
11    }  
12  }  
13 }
```